On the Choice of Random Directions for Stochastic Approximation Algorithms

James Theiler and Jarod Alper

This work was supported by the Laboratory Directed Research and Development (LDRD) program at Los Alamos National Laboratory.

J. Theiler is a technical staff member in the Space and Remote Sensing Sciences Group, MS-B244, Los Alamos National Laboratory, Los Alamos, NM 87545. Email: jt@lanl.gov (Phone: 505-665-5682; fax: 505-665-4414)

J. Alper recently graduated from Brown University with a degree in Computer Science; he was graduate research assistant at Los Alamos when this work was done. Email: Jarod_Alper@alumni.brown.edu.

Abstract

We investigate variants of the Kushner-Clark Random Direction Stochastic Approximation (RDSA) algorithm for optimizing noisy loss functions in high-dimensional spaces. These variants employ different strategies for choosing random directions. The most popular approach is random selection from a Bernoulli distribution, which for historical reasons goes also by the name Simultaneous Perturbation Stochastic Approximation (SPSA). But viable alternatives include an axis-aligned distribution, a normal distribution, and a uniform distribution on a spherical shell. Although there are special cases where the Bernoulli distribution is optimal, there are other cases where it performs worse than other alternatives. We find that for generic loss functions that are not aligned to the coordinate axes, the average asymptotic performance is depends only on the *radial* fourth moment of the distribution of directions, and is identical for Bernoulli, the axis-aligned, and the spherical shell distributions. Of these variants, the spherical shell is optimal in the sense of minimum variance over random orientations of the loss function with respect to the coordinate axes. We also show that for unaligned loss functions, the performance of the Keifer-Wolfowitz-Blum Finite Difference Stochastic Approximation (FDSA) is asymptotically equivalent to the RDSA algorithms, and we observe numerically that the pre-asymptotic performance of FDSA is often superior. We also introduce a "quasirandom" selection process which exhibits the same asymptotic performance, but empirically is observed to converge to the asymptote more rapidly.

Index Terms

stochastic approximation, optimization, noisy loss function, random direction, finite difference, simultaneous perturbation

I. INTRODUCTION

Stochastic approximation provides a simple and effective approach for finding roots and minima of functions whose evaluations are contaminated with noise. Consider a smooth[1] p-dimensional loss function $L : \mathbb{R}^p \to \mathbb{R}$, with gradient $\mathbf{g} : \mathbb{R}^p \to \mathbb{R}^p$. Assume that L has a unique[2] local (and therefore global) minimum $\mathbf{x}^* \in \mathbb{R}^p$. That is, $L(\mathbf{x}^*) \leq L(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^p$, and $\mathbf{g}(\mathbf{x}) = 0$ iff $\mathbf{x} = \mathbf{x}^*$.

If a direct (but possibly noisy) estimator $\hat{\mathbf{g}}(\mathbf{x})$ of the gradient function is available, then the Robbins-Monro [1] algorithm (as extended by Blum [2] to multidimensional systetms) estimates

[1]To simplify exposition, we take L to be infinitely differentiable, but remark that many of the results only require that L be s-times differentiable, where s depends on the particular result.

[2]Stochastic approximation algorithms can still be useful for loss functions with multiple local minima, but formal results are more readily obtained if there is a single local minimum.

a root of $\mathbf{g}(\mathbf{x})$ with the following recursion:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - a_k \hat{\mathbf{g}}(\mathbf{x}_k), \tag{1}$$

where a_k is a sequence of positive numbers that satisfy $\sum_{k=1}^{\infty} a_k = \infty$ and $\lim_{k\to\infty} a_k = 0$. In particular, $a_k = a_o/k^\alpha$ with $0 < \alpha \leq 1$ satisfies these conditions. If the estimator is unbiased, that is $E\{\, \hat{\mathbf{g}}(\mathbf{x}) \,\} = \mathbf{g}(\mathbf{x})$, then $\mathbf{x}_k$ will converge to the root of $\mathbf{g}$. In particular, it can be shown that $E\{\, (\mathbf{x}_k - \mathbf{x}^*)^2 \,\} = O(k^{-\alpha})$ for large k.

Kiefer and Wolfowitz [3] introduced an algorithm in which the gradient is estimated by finite differences (Blum [2] also extended this result to multiple dimensions). This finite difference stochastic approximation (FDSA) algorithm employs an estimator for the gradient whose ith component is given by

$$\hat{g}_i(\mathbf{x}) = \frac{\hat{L}(\mathbf{x} + c\mathbf{e}_i) - \hat{L}(\mathbf{x} - c\mathbf{e}_i)}{2c}, \tag{2}$$

where $\mathbf{e}_i$ is the unit vector along the ith axis, and $\hat{L}$ is a noisy measurement of the loss function. Since this is done for each component, it requires $2p$ measurements of the loss function for each iteration. For $c > 0$, Eq. (2) is in general a biased estimator of the gradient. Convergenc is achieved by providing a decreasing sequence c_k with $\lim_{n\to\infty} c_k = 0$ so that the bias is eventually eliminated. However, the cost of using a smaller c is a larger variance, so the rate at which $c_k \to 0$ must be carefully chosen.

In the FDSA algorithm, separate estimates are computed for each component of the gradient. This means that a p-dimensional problem requires at least $2p$ evaluations of the loss function per iteration. By contrast, the random direction stochastic approximation (RDSA) algorithms estimate only one component of the gradient per iteration. Let $\boldsymbol{\xi} \in \mathbb{R}^p$ be a direction vector. In Kushner and Clark [4], $\boldsymbol{\xi}$ is treated as a unit vector with $|\boldsymbol{\xi}|^2 = \sum_i \xi_i^2 = 1$ and since it is a random direction, it satisfies $E\Big\{\, \boldsymbol{\xi}\boldsymbol{\xi}^T \,\Big\} = I/p$. Chin [5] prefers the convention that $\boldsymbol{\xi}$ have radius[3] $\sqrt{p}$ so that $|\boldsymbol{\xi}|^2 = \sum_i \xi_i^2 = p$ and $E\Big\{\, \boldsymbol{\xi}\boldsymbol{\xi}^T \,\Big\} = I$. Regardless of convention for $\boldsymbol{\xi}$, both authors write the RDSA formula as

$$\mathbf{x}_{n+1} = \mathbf{x}_k - a_k \left[\frac{\hat{L}(\mathbf{x}_k + c_k\boldsymbol{\xi}_k) - \hat{L}(\mathbf{x}_k - c_k\boldsymbol{\xi}_k)}{2c_k} \right] \boldsymbol{\xi}_k, \tag{3}$$

but it bears remarking that the formulas are not equivalent. Using the $|\boldsymbol{\xi}|^2 = p$ convention, the above formula corresponds directly to the Robbins-Monro formulation in Eq. (1). With the

[3] Chin [5] mistakenly says the radius is p.

$|\boldsymbol{\xi}|^2 = 1$ convention, however, the above formula corresponds to $\mathbf{x}_{n+1} = \mathbf{x}_k - (1/p)a_k\hat{\mathbf{g}}(\mathbf{x}_k)$, which by a simple rescaling of a_n is equivalent[4] to Eq. (1). To facilitate comparisons with the more recent work, we will take the convention that $|\boldsymbol{\xi}|^2 = p$.

Several choices are available for choosing the random distributions.

- The *Axis* distribution is the simplest: $\boldsymbol{\xi} = \pm\sqrt{p}\mathbf{e}_i$ with coordinate i chosen at random from $\{1, \ldots, p\}$.
- The *Normal* distribution distribution is obtained by taking each component ξ_i to be distributed as $N(0, 1)$. The normal distribution is fully isotropic, but its radius is not fixed. The average squared radius, however, is the same as for the other distributions: $E\{\, |\boldsymbol{\xi}|^2 \,\} = p$.
- The *Shell* distribution, like the normal, is also fully isotropic, but its radius is fixed. In practice, this is achieved by taking a vector from the normal distribution and then rescaling it so that the radius is exactly $\sqrt{p}$.
- Finally, the *Bernoulli* distribution is obtained by taking each component ξ_i at random from $\{-1, 1\}$.

Note that in the RDSA algorithm, the gradient is estimated in the direction $\boldsymbol{\xi}$ and the adjustment to $\mathbf{x}_k$ is in the same direction $\boldsymbol{\xi}$. By contrast, the general form of the SPSA algorithm introduced by Spall [6] employs two different directions, $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$, and the estimate is given by

$$\hat{\mathbf{g}}(\mathbf{x}) = \left[\frac{\hat{L}(\mathbf{x} + c\boldsymbol{\xi}) - \hat{L}(\mathbf{x} - c\boldsymbol{\xi})}{2c}\right] \boldsymbol{\zeta}, \tag{4}$$

where $\boldsymbol{\xi}$ is chosen from a distribution that has to satisfy some particular constraints, and the components of $\boldsymbol{\zeta}$ are given by

$$\zeta_i = 1/\xi_i. \tag{5}$$

Only in the special case of a Bernoulli distribution do we have the geometrically plausible $\boldsymbol{\xi} = \boldsymbol{\zeta}$. Sadegh and Spall [7] have shown that the Bernoulli distribution is always the best choice for SPSA, and in fact the Bernoulli distribution is the only choice that has ever been advocated for SPSA. Thus, SPSA is really just a special case of RDSA, though it does bear remarking – to

[4] In a footnote, Chin [5, p.245] claims that it is "incorrect" to use a radius of 1, and says that Kushner and Clark [4, p.60] are mistaken in their proof of convergence. Our reading of Kushner and Clark (see especially the remark on p.59: "except that (2.3.18) replaces (2.3.11)") suggests that they have correctly accounted for this extra factor of p.

the credit of SPSA's inventor – that the use of a Bernoulli distribution with RDSA had not been suggested until after SPSA had been introduced.

Because the optimal direction choice for SPSA is the Bernoulli distribution, there is some sense that the Bernoulli distribution might be optimal for RDSA as well. Chin [5] provides both analytic arguments and results of a numerical experiment to show that SPSA (that is, RDSA with Bernoulli directions) outperforms RDSA (with a normal distribution). Some of Chin's findings have been repeated elsewhere in the literature. For instance, Wang and Chong [8] note that Bernoulli is optimal for SPSA, and go on to suggest that by following the approach of Sadegh and Spall [7], they "can show that the Bernoulli distribution is also optimal" for the RDSA algorithm, "based on the asymptotic distribution established by Chin" in Ref. [5].

Chin [5] also includes a comparison with FDSA and finds that the FDSA algorithm requires p times as much computation to achieve the same result. This is another result that has been widely reported, and in fact Spall [9], [10], [11], [12] claims that:

> Under reasonably general conditions, SPSA and Kiefer-Wolfowitz finite-difference-based SA (FDSA) achieve the same level of statistical accuracy for a given number of iterations even though SPSA uses p times fewer evaluations than FDSA (since each gradient approximation uses only $1/p$ the number of function evaluations).[5]

However, we will argue that the "reasonably general" conditions for achieving this full factor of p are in fact rather special, and we question whether one should reasonably expect to encounter these conditions in practice.

II. Convergence Rates for Stochastic Approximation Algorithms

To achieve an optimal convergence rate, there is a trade-off between variance and bias. We can write the gradient estimator as a sum of three terms: the true gradient $\mathbf{g}$, the noise $\boldsymbol{\eta}$, and the bias $\mathbf{b}$:

$$\hat{\mathbf{g}}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) + \boldsymbol{\eta}(\mathbf{x}) + \mathbf{b}(\mathbf{x}), \tag{6}$$

where

$$\boldsymbol{\eta}(\mathbf{x}) = \hat{\mathbf{g}}(\mathbf{x}) - E\{\,\hat{\mathbf{g}}(\mathbf{x})\,\}, \tag{7}$$

$$\mathbf{b}(\mathbf{x}) = E\{\,\hat{\mathbf{g}}(\mathbf{x})\,\} - \mathbf{g}(\mathbf{x}). \tag{8}$$

[5] Although Spall makes this statement (verbatim) in the three references and on his website, the source that is cited (his own 1992 paper [6]) does not make such a sweeping statement.

In general, the more accurately the gradient $\mathbf{g}(\mathbf{x})$ is estimated, the more rapidly the iteration in Eq. (1) will converge to the solution. In the following two subsections, we will derive first the variance and then the bias for different estimators of the gradient, and in the subsection after that, we will show how these relate to the convergence of Eq. (1).

A. Variance

From the definition of noise in Eq. (7), we can write for the FDSA case

$$\eta_i = \frac{\epsilon_+ - \epsilon_-}{2c}, \tag{9}$$

where $\epsilon_\pm = \hat{L}(\mathbf{x} \pm c\mathbf{e}_i) - L(\mathbf{x} \pm c\mathbf{e}_i)$ is the noise in the measurement of the loss function; it has mean zero and variance σ^2. The covariance of the noise is given by

$$E\Big\{ \boldsymbol{\eta}\boldsymbol{\eta}^T \Big\} = \frac{\sigma^2}{2c^2} I, \tag{10}$$

and the total variance in the noise is the trace of the covariance matrix:

$$E\Big\{ \boldsymbol{\eta}^T\boldsymbol{\eta} \Big\} = \text{tr}\left(E\Big\{ \boldsymbol{\eta}\boldsymbol{\eta}^T \Big\} \right) = \frac{p\sigma^2}{2c^2}. \tag{11}$$

For RDSA, we do not compute each component separately, so the total noise is given by

$$\boldsymbol{\eta} = \frac{\epsilon_+ - \epsilon_-}{2c}\boldsymbol{\xi}, \tag{12}$$

and the covariance and variance are given by

$$E\Big\{ \boldsymbol{\eta}\boldsymbol{\eta}^T \Big\} = \frac{\sigma^2}{2c^2} E\Big\{ \boldsymbol{\xi}\boldsymbol{\xi}^T \Big\} = \frac{\sigma^2}{2c^2} I, \tag{13}$$

$$E\Big\{ \boldsymbol{\eta}^T\boldsymbol{\eta} \Big\} = \text{tr}\left(E\Big\{ \boldsymbol{\eta}\boldsymbol{\eta}^T \Big\} \right) = \frac{p\sigma^2}{2c^2}, \tag{14}$$

which is the same as the FDSA case, and is the same regardless of which RDSA distribution is used.

B. Bias

Although we saw that the variance in the estimator of the gradient did not depend on choice of RDSA distribution, it will turn out that the *bias* in the estimator can depend on this choice.

The bias is defined in Eq. (8), and for RDSA, we can write this

$$\begin{aligned}\mathbf{b}(\mathbf{x}) &= E\Big\{ \frac{L(\mathbf{x}+c\boldsymbol{\xi}) - L(\mathbf{x}-c\boldsymbol{\xi})}{2c}\,\boldsymbol{\xi} \Big\} - \mathbf{g}(\mathbf{x}) \\ &= E\Big\{ L_\xi^{(1)}(\mathbf{x})\,\boldsymbol{\xi} \Big\} + \frac{1}{6}\,c^2 E\Big\{ L_\xi^{(3)}(\mathbf{x})\,\boldsymbol{\xi} \Big\} + O(c^4) - \mathbf{g}(\mathbf{x}), \end{aligned} \tag{15}$$

where the scalar $L_\xi^{(n)}(\mathbf{x})$ is the nth derivative of L in the direction $\boldsymbol{\xi}$; it is defined by

$$L_\xi^{(n)}(\mathbf{x}) = \frac{\partial^n}{\partial t^n} L(\mathbf{x}+\boldsymbol{\xi}t). \tag{16}$$

In particular, $L_\xi^{(1)}(\mathbf{x}) = \sum_i g_i(\mathbf{x})\xi_i$, and

$$L_\xi^{(3)}(\mathbf{x}) = \sum_{ijk} L_{ijk}(\mathbf{x})\xi_i\xi_j\xi_k, \tag{17}$$

where L_{ijk} is third derivative along the axes $\mathbf{e}_i$, $\mathbf{e}_j$ and $\mathbf{e}_k$.

We will assume that we are in the asymptotic regime, where $\mathbf{x} \to \mathbf{x}^*$, and c is small, so we can neglect the $O(c^4)$ terms. Then, the ℓth component of the bias $\mathbf{b}$ is given by

$$\begin{aligned} b_\ell &= E\Big\{ (\sum_i g_i\xi_i)\,\xi_\ell \Big\} - g_\ell + \frac{1}{6}\,c^2 \sum_{ijk} L_{ijk} E\{\,\xi_i\xi_j\xi_k\xi_\ell\,\} \\ &= \sum_i g_i E\{\,\xi_i\xi_\ell\,\} - g_\ell + \frac{1}{6}\,c^2 \sum_{ijk} L_{ijk} E\{\,\xi_i\xi_j\xi_k\xi_\ell\,\}. \end{aligned} \tag{18}$$

For nearly all random direction schemes that might be considered, and certainly for the schemes considered in this paper, the only nonzero values for $E\{\,\xi_i\xi_\ell\,\}$ and $E\{\,\xi_i\xi_j\xi_k\xi_\ell\,\}$ are those that involve even moments. We have

$$b_\ell = g_\ell(E\Big\{\,\xi_\ell^2\,\Big\} - 1) + \frac{1}{6}\,c^2 \left[L_{\ell\ell\ell} E\Big\{\,\xi_\ell^4\,\Big\} + 3 \sum_{i=1, i\neq\ell}^{p} L_{ii\ell} E\Big\{\,\xi_\ell^2\xi_i^2\,\Big\} \right], \tag{19}$$

where the factor of three comes from the equality of $L_{ii\ell} = L_{i\ell i} = L_{\ell ii}$ for smooth functions. Furthermore, $E\{\,\xi_\ell^2\,\} = 1$ (since $E\{\,\sum_i \xi_i^2\,\} = p$), so the first order gradient term vanishes. We will define

$$\tau \equiv E\Big\{\,\xi_\ell^4\,\Big\}, \tag{20}$$

$$\nu \equiv E\Big\{\,\xi_\ell^2\xi_i^2\,\Big\}, \tag{21}$$

so then the bias can be expressed as

$$b_\ell = \frac{1}{6}\,c^2 \left[\tau L_{\ell\ell\ell} + 3\nu \sum_{i=1, i\neq\ell}^{p} L_{ii\ell} \right]. \tag{22}$$

	$\tau = E\{\, \xi_\ell^4 \,\}$	$\nu = E\{\, \xi_\ell^2 \xi_m^2 \,\}$	$p\tau + p(p-1)\nu$	$[\tau - 3\nu]^2$
Distribution				
Bernoulli	1	1	p^2	4
Axis	p	0	p^2	p^2
Normal	3	1	$p(p+2)$	0
Shell	$3p/(p+2)$	$p/(p+2)$	p^2	0

TABLE I

FOUR RANDOM DIRECTION DISTRIBUTIONS AND STATISTICS THAT CHARACTERIZE THE BIAS IN THEIR ESTIMATES OF GRADIENT.

Note that Eq. (2.4) in Chin [5] gives the incorrect form $b_\ell = \frac{1}{6}\, c^2 \tau \left[L_{\ell\ell\ell} + 3 \sum_{i=1, i\neq\ell}^{p} L_{ii\ell} \right]$. As it turns out, Chin's formula *is* correct for the Bernoulli distribution, for in that case $\tau = \nu$. Chin's formula is also correct if all the cross-derivatives (L_{ijk} except when $i = j = k$) happen to be zero, but in that degenerate case a much simpler expression would be used. To compare different random direction distributions on general loss functions, the correct expression in Eq. (22) should be employed.

Table I shows the values for τ and ν for different random direction distributions. The computation is straightforward for Bernoulli, Axis, and Normal; for Shell, the derivation is given in the Appendix I.

For the FDSA algorithm, the bias is given by

$$b_\ell = \frac{1}{6}\, c^2 L_{\ell\ell\ell}. \tag{23}$$

For the same value of c, this is generally much smaller than the RDSA bias, but the price paid is p times as many evaluations of the loss function. If the cross-derivatives are zero, however, then the more expensive FDSA estimator has the same bias as the RDSA-Bernoulli estimator.

C. Convergence of stochastic approximation

Having derived the variance and bias for various estimators $\hat{\mathbf{g}}(\mathbf{x})$ of the gradient, we will now show how these are related to the convergence $\mathbf{x}_k \to \mathbf{x}^*$. We invoke a theorem of Fabian [13],

regarding recursions of the form

$$\mathbf{u}_{k+1} = (I - k^{-\alpha}\Gamma_k)\mathbf{u}_k + k^{-(\alpha+\beta)/2}\Phi_k\mathbf{v}_k + k^{-\alpha-\beta/2}\mathbf{t}_k, \tag{24}$$

where $\Gamma_k \to \Gamma$ and $\Phi_k \to \Phi$ are $p \times p$ matrices, $\mathbf{v}_k$ is a noise vector for which $E\{\,\mathbf{v}_k\,\} = 0$ and $E\Big\{\,\mathbf{v}_k\mathbf{v}_k^T\,\Big\} \to \Sigma$ where Σ is a $p \times p$ covariance matrix, and $\mathbf{t}_k \to \mathbf{t}$ is a p-dimensional vector. Such recurrences converge to zero at a rate $\mathbf{u}_k \sim k^{-\beta/2}$ and in particular, converge asymptotically to a scaled gaussian with a specific mean $\boldsymbol{\mu}$ and covariance $\mathbf{V}$:

$$\lim_{k\to\infty} k^{\beta/2}u_k = \mathcal{N}(\boldsymbol{\mu}, \mathbf{V}). \tag{25}$$

We refer the reader to Ref. [13] for more precise details about the conditions of convergence and for the specific expressions for mean and covariance.

Note that we can write the true gradient $\mathbf{g}(\mathbf{x})$ in terms of the Hessian $\mathbf{H}(\mathbf{x})$, where $H_{ij} = \partial^2 L/\partial x_i \partial x_j$:

$$\mathbf{g}(\mathbf{x}) = \mathbf{H}(\mathbf{x}')\,(\mathbf{x} - \mathbf{x}^*) \tag{26}$$

for a point $\mathbf{x}'$ that is somewhere on the segment between $\mathbf{x}$ and $\mathbf{x}^*$. Thus, we can rewrite the Robbins-Monro recursion in Eq. (1) as

$$\mathbf{x}_{k+1} - \mathbf{x}^* = [I - a_k\mathbf{H}(\mathbf{x}'_k)]\,(\mathbf{x}_k - \mathbf{x}^*) - a_k\mathbf{b}(\mathbf{x}_k) - a_k\boldsymbol{\eta}(\mathbf{x}_k). \tag{27}$$

We take $a_k = a_o k^{-\alpha}$ and make the associations

$$\begin{aligned}
\mathbf{u}_k &= \mathbf{x}_k - \mathbf{x}^*, && (28)\\
\Gamma_k &= a_o\mathbf{H}(\mathbf{x}'_k), && (29)\\
\Phi_k &= I, && (30)\\
\mathbf{v}_k &= -a_o k^{(-\alpha+\beta)/2}\boldsymbol{\eta}(\mathbf{x}_k), && (31)\\
\mathbf{t}_k &= -a_o k^{\beta/2}\mathbf{b}(\mathbf{x}_k). && (32)
\end{aligned}$$

The boundedness of the sequences $\mathbf{v}_k$ and $\mathbf{t}_k$ provide conditions on β in terms of α and γ. In particular, $E\Big\{\,\mathbf{v}_k\mathbf{v}_k^T\,\Big\} \to \Sigma$ implies that $a_o^2 k^{-\alpha+\beta}\boldsymbol{\eta}\boldsymbol{\eta}^T \to \Sigma$. From Eq. (13) and the schedule $c_k = c_o k^{-\gamma}$ we have

$$\frac{a_o^2\sigma^2}{2c_o^2}k^{-\alpha+\beta+2\gamma} \to \Sigma, \tag{33}$$

which imposes the condition $-\alpha + \beta + 2\gamma \leq 0$ or

$$\beta \leq \alpha - 2\gamma. \tag{34}$$

We have found that the bias depends on the details of the gradient estimator, but the general form is

$$\mathbf{b}(\mathbf{x}) = \tilde{\mathbf{b}}(\mathbf{x})c^{2s} = \tilde{\mathbf{b}}(\mathbf{x})c_o^{2s}k^{-2s\gamma}, \tag{35}$$

where $\tilde{\mathbf{b}}(\mathbf{x})$ depends both on the details of the loss function and the variant of RDSA or FDSA that is used, but does not depend on c and therefore has a well-defined limit $\tilde{\mathbf{b}} = \lim_{k\to\infty} \tilde{\mathbf{b}}(\mathbf{x}_k)$. Also, $s = 1$ for the usual two-point estimators of gradient, but can be larger than one for the higher-order estimators described in Appendix II. The condition $\mathbf{t}_k \to \mathbf{t}$ leads to

$$-a_o c_o^{2s}\tilde{\mathbf{b}}(\mathbf{x}_k)k^{\beta/2-2s\gamma} \to \mathbf{t}, \tag{36}$$

which implies $\beta/2 - 2s\gamma \leq 0$ or

$$\beta \leq 4s\gamma. \tag{37}$$

Combining Eqs. (25,34,37), we have the convergence given by $E\{\,|\mathbf{x}_k - \mathbf{x}^*|^2\,\} = O(k^{-\beta})$ where $\beta \leq \min(\alpha - 2\gamma, 4s\gamma)$. The optimal convergence maximizes β, and this occurs when $\alpha = 1$ and $\gamma = 1/(4s+2)$, leading to $\beta = 2s/(2s+1)$. In particular, for the usual $s = 1$ case, we have $\gamma = 1/6$ and $\beta = 2/3$.

With only slight loss of generality (and a huge savings in the amount of tedious bookkeeping we have to keep track of), we will assume that the data have been scaled so that the Hessian is a multiple of the identity matrix. In fact, we will take $H(\mathbf{x}^*) = 2I$, which is equivalent to assuming that the loss function is of the form $L(\mathbf{x}) = |\mathbf{x} - \mathbf{x}^*|^2 +$ higher order terms.

For the case of optimal α, β, and γ, and with the simplified Hessian, Fabian's result in Ref. [13] leads to a simplified expression

$$\lim_{k\to\infty} E\Big\{\, k^{\beta}|\mathbf{x}_k - \mathbf{x}^*|^2 \,\Big\} = \frac{a_o^2 c_o^{-2} p\sigma^2}{2a_o - \beta/2} + \frac{a_o^2 c_o^{4s}|\tilde{\mathbf{b}}|^2}{(2a_o - \beta/2)^2}. \tag{38}$$

We remark that the first term involves the noise in the estimator of the loss function, and that is essentially the same for FDSA and the different RDSA variants. However, the second term, which is proportional to the bias in the estimator of the gradient, does depend on those differences, and that is the focus of our comparisons.

This expression allows us to optimize the parameters a_o and c_o. By taking the derivative with resepect to c_o and setting to zero, we can get a general sense of scaling for the optimal c_o:

$$c_o \sim \left(\frac{p\sigma^2}{|\tilde{\mathbf{b}}|^2} \right)^{1/(4s+2)}. \tag{39}$$

And specifically for the standard case $s = 1$, we have $a_o = 1/2$, and

$$c_o = \left(\frac{2p\sigma^2}{3|\tilde{\mathbf{b}}|^2} \right)^{1/6}, \tag{40}$$

and putting this back into Eq. (38) leads to

$$E\Big\{ |\mathbf{x}_k - \mathbf{x}^*|^2 \Big\} \sim k^{-2/3} |\tilde{\mathbf{b}}|^{2/3} p^{2/3} \sigma^{4/3}. \tag{41}$$

For RDSA, there are $N = 2k$ function evaluations in k iterations, so the mean squared error scales as

$$\text{MSE}_{\text{RDSA}} \sim N^{-2/3} |\tilde{\mathbf{b}}|^{2/3} p^{2/3} \sigma^{4/3}, \tag{42}$$

while for FDSA, there are $N = 2pk$ evaluations by the kth iteration,so

$$\text{MSE}_{\text{FDSA}} \sim N^{-2/3} |\tilde{\mathbf{b}}|^{2/3} p^{4/3} \sigma^{4/3}. \tag{43}$$

A natural measure of the relative "cost" of the different algorithms is the number N of loss function evaluations required to achieve a given level of accuracy. If we let ϵ be the desired accuracy, then then we can rewrite the previous two equations as

$$\text{N}_{\text{RDSA}} \sim |\tilde{\mathbf{b}}| p \sigma^2 \epsilon^{-3} \tag{44}$$

and

$$\text{N}_{\text{FDSA}} \sim |\tilde{\mathbf{b}}| p^2 \sigma^2 \epsilon^{-3}. \tag{45}$$

D. Expected bias for generically rotated loss functions

The bias in an estimator of gradient for a given loss function depends on the relative orientation of the loss function with respect to the coordinate axes. Fig. 3 provides a clear visualization of this concept; here, a forty-five degree rotation would swap the Axis directions with the Bernoulli directions.

To better understand this effect, we consider loss functions with "generic" orientations with respect to the coordinate axes. Conceptually, we do this by randomly rotating the loss function. But a direct average of the bias over random rotations of the loss function would produce a zero. So instead, we randomly rotate the coordinate axes with respect to which the random directions are chosen. Suppose $\boldsymbol{\xi}$ is chosen from some distribution $P(\boldsymbol{\xi})$. Then we will consider a new distribution $P_U(\boldsymbol{\xi}) = P(U\boldsymbol{\xi})$ which is the distribution $P(\boldsymbol{\xi})$ rotated according to the orthogonal matrix U.

We will compute τ_U, the fourth moment of ξ'_ℓ where $\boldsymbol{\xi}'$ is taken from the distribution P_U.

$$\begin{aligned}
\tau_U &= E\Big\{ (\xi'_\ell)^4 \Big\} = E\Big\{ (U\xi)^4_\ell \Big\} \\
&= E\left\{ \sum_{ijkm} u_{i\ell} u_{j\ell} u_{k\ell} u_{m\ell} \xi_i \xi_j \xi_k \xi_m \right\} \\
&= \sum_{ijkm} u_{i\ell} u_{j\ell} u_{k\ell} u_{m\ell} E\{ \xi_i \xi_j \xi_k \xi_m \}. \qquad (46)
\end{aligned}$$

The average over all random directions of $E\{ \xi_i \xi_j \xi_k \xi_m \}$ is given by

$$\begin{aligned}
E\{ \xi_i \xi_j \xi_k \xi_m \} = \quad & \tau \delta(i=j=k=m) \\
+ \quad & \nu \delta((i=j) \neq (k=m)) \\
+ \quad & \nu \delta((i=k) \neq (m=j)) \\
+ \quad & \nu \delta((i=m) \neq (j=k)), \qquad (47)
\end{aligned}$$

and therefore

$$\tau_U = \tau \sum_i u_{i\ell}^4 + 3\nu \sum_{i \neq j} u_{i\ell}^2 u_{j\ell}^2. \qquad (48)$$

For a rotated loss function, the bias would be proportional to τ_U instead of the τ that is appropriate for the unrotated loss function. If we consider an ensemble of rotated loss functions, the average τ_U would be given by[6]

$$\begin{aligned}
\langle \tau_U \rangle &= \tau \left\langle \sum_i u_{i\ell}^4 \right\rangle + 3\nu \left\langle \sum_{i \neq j} u_{i\ell}^2 u_{j\ell}^2 \right\rangle \\
&= \tau p \left\langle u_{i\ell}^4 \right\rangle + 3\nu p(p-1) \left\langle u_{i\ell}^2 u_{j\ell}^2 \right\rangle. \qquad (49)
\end{aligned}$$

[6] So far, we have done two kinds of averaging, and now this is a third. We use $E\{ \bullet \}$ to indicate expectation over loss function evaluation, and to indicate averages over random directions. Now, we are using $\langle \bullet \rangle$ to represent an average over random rotation matrices U.

Since the matrix elements $u_{\bullet\ell}$ form orthogonal vectors of unit length, we can treat them as axes of a p-dimensional sphere, and compute moments as described in Appendix I by Eqs. (88-93). In particular, we have

$$\begin{aligned}\langle \tau_U \rangle &= \tau p \, \frac{3}{p(p+2)} + 3\nu p(p-1) \, \frac{1}{p(p+2)} \\ &= \frac{3}{p(p+2)} \left[p\tau + p(p-1)\nu \right]. \end{aligned} \tag{50}$$

A similar argument shows

$$\langle \nu_U \rangle = \frac{1}{p(p+2)} \left[p\tau + p(p-1)\nu \right]. \tag{51}$$

But it is important to note that

$$p\tau + p(p-1)\nu = pE\Big\{ \xi_i^4 \Big\} + p(p-1)E\Big\{ \xi_i^2 \xi_j^2 \Big\} \tag{52}$$

$$= E\left\{ (\sum_i \xi_i^2)^2 \right\} = E\Big\{ r^4 \Big\}, \tag{53}$$

where $E\{ r^4 \}$ is the *radial* fourth moment of the direction $\boldsymbol{\xi}$. Since the radius is normalized so that $r = \sqrt{p}$, we have $E\{ r^4 \} = p^2$ for the Bernoulli, Axis, and Shell distributions; for the Normal distribution, the radius is only normalized on average, and we have $E\{ r^4 \} = p(p+2)$.

Since the averages of τ_U and ν_U depend only on the radial fourth moment, they are the same for Axis, Bernoulli, and Shell. So for generic orientations of the loss functions with respect to the coordinate axes, all three random direction distributions will on average give the same result. In particular, we have that the rotation-averaged bias is given by

$$\begin{aligned}\langle b_\ell \rangle &= \frac{1}{6} c^2 \left[\frac{p\tau + p(p-1)\nu}{p(p+2)} \right] \left(3L_{\ell\ell\ell} + 3 \sum_{i=1, i\neq\ell}^{p} L_{ii\ell} \right) \\ &= \frac{1}{2} c^2 \left[\frac{E\{ r^4 \}}{p(p+2)} \right] \sum_{i=1}^{p} L_{ii\ell}. \end{aligned} \tag{54}$$

Note that for the FDSA algorithm, the rotation-averaged bias can be shown to be

$$\langle b_\ell \rangle = \frac{1}{2} c^2 \left[\frac{1}{p+2} \right] \sum_{i=1}^{p} L_{ii\ell}, \tag{55}$$

which is a factor of p smaller than the RDSA bias in Eq. (54) for the Bernoulli, Axis, and Shell distributions. Thus, based on Eq. (44) and Eq. (45), we see that the asymptotic efficiency of FDSA and RDSA (except for the Normal random direction distribution) are on average the same for loss functions which do not exhibit any special alignment to their coordinate axes.

E. Squared bias

We have seen that the mean (averaged over all orientations) bias is identical for Axis, Bernoulli, and Shell. But for any given orientation, one may well be better than another, as we have seen. A question that then arises regarding the reliability of different methods, and one way to measure this is by averaging the squared bias over all orientations. If two methods have the same mean bias, but one method has a bias that varies from small to large as the orientation changes while the other has a more consistent bias, then the first method with have a larger squared bias. For a loss function that is completly symmetric under rotation, the magnitude of bias will be the same for all rotations, regardless of the method used for choosing random directions. So the squared bias also depends on the anisotropy of the loss function.

We compute the rotation-averaged squared bias:

$$\begin{aligned}
\left\langle \mathbf{b}^T\mathbf{b} \right\rangle &= \sum_{\ell=1}^{p} \left\langle b_\ell^2 \right\rangle \\
&= \frac{1}{36}c^4 \sum_{\ell=1}^{p} \left\langle \left(\tau L_{\ell\ell\ell} + 3\nu \sum_{m\neq\ell} L_{\ell mm} \right)^2 \right\rangle \qquad (56) \\
&= \frac{1}{36}c^4 \Big[\tau^2 p \left\langle L_{\ell\ell\ell}^2 \right\rangle + 6\tau\nu p(p-1) \left\langle L_{\ell\ell\ell} L_{\ell mm} \right\rangle \\
&\quad + 9\nu^2 p(p-1) \left\langle L_{\ell mm}^2 \right\rangle + 9\nu^2 p(p-1)(p-2) \left\langle L_{\ell mm} L_{\ell nn} \right\rangle \Big]. \qquad (57)
\end{aligned}$$

Although the average value of $L_{\ell mn}$ – averaged over rotations specified by random orthogonal matrices U – is zero for any choice of ℓmn, the averages of products of these third derivatives will generally be nonzero. In general, we can use

$$L_{\ell mn} = \sum_{ijk} u_{i\ell} u_{jm} u_{kn} L'_{ijk} \qquad (58)$$

to describe the rotated third derivative $L_{\ell mn}$ in terms of the original third derivatives L'_{ijk}.

Forgoing full generality, we will consider loss functions whose anisotropy is characterized, like that of the Chin function (described in the next section), by the existence of a coordinate system in which the loss is a sum of separate but identical losses on each of the coordinates. That is:

$$L(\mathbf{z}) = \sum_i \mathcal{L}(\sum_j u_{ij} z_j), \qquad (59)$$

where u_{ij} are the matrix elements of the rotation matrix U. And the third derivative looks like

$$L_{\ell mn} = \mathcal{L}''' \sum_i u_{i\ell} u_{im} u_{in}. \tag{60}$$

To compute the expectation value for the various products that appear in Eq. (57), we substitute from Eq. (60):

$$\left\langle L^2_{\ell\ell\ell} \right\rangle / (\mathcal{L}''')^2 = \sum_{jk} \left\langle u^3_{j\ell} u^3_{k\ell} \right\rangle = \sum_j \left\langle u^6_{j\ell} \right\rangle = p \left\langle u^6_{j\ell} \right\rangle, \tag{61}$$

and similarly,

$$\left\langle L^2_{\ell mm} \right\rangle / (\mathcal{L}''')^2 = p \left\langle u^2_{j\ell} u^4_{jm} \right\rangle, \tag{62}$$

$$\langle L_{\ell\ell\ell} L_{\ell mm} \rangle / (\mathcal{L}''')^2 = p \left\langle u^4_{j\ell} u^2_{jm} \right\rangle, \tag{63}$$

$$\langle L_{\ell mm} L_{\ell nn} \rangle / (\mathcal{L}''')^2 = p \left\langle u^2_{j\ell} u^2_{jm} u^2_{jn} \right\rangle. \tag{64}$$

The matrix elements form orthogonal vectors of unit length, so the results of Appendix I can be used to obtain:

$$\left\langle L^2_{\ell\ell\ell} \right\rangle / (\mathcal{L}''')^2 = \frac{15}{(p+2)(p+4)}, \tag{65}$$

$$\left\langle L^2_{\ell mm} \right\rangle / (\mathcal{L}''')^2 = \frac{3}{(p+2)(p+4)}, \tag{66}$$

$$\langle L_{\ell\ell\ell} L_{\ell mm} \rangle / (\mathcal{L}''')^2 = \frac{3}{(p+2)(p+4)}, \tag{67}$$

$$\langle L_{\ell mm} L_{\ell nn} \rangle / (\mathcal{L}''')^2 = \frac{1}{(p+2)(p+4)}. \tag{68}$$

Substituting into Eq. (57),

$$\begin{aligned} \left\langle \mathbf{b}^T \mathbf{b} \right\rangle &= \frac{p(\mathcal{L}''')^2 c^4}{36(p+2)(p+4)} \Big[15\tau^2 \, + \, 18\tau\nu(p-1) \\ &\quad + 27\nu^2(p-1) \, + \, 9\nu^2(p-1)(p-2) \Big], \end{aligned} \tag{69}$$

and with a little algebraic manipulation, we can obtain

$$\begin{aligned} \left\langle \mathbf{b}^T \mathbf{b} \right\rangle &= p \left(\frac{\mathcal{L}''' c^2}{2p(p+2)} [p\tau + p(p-1)\nu] \right)^2 \\ &\quad + \frac{p(p-1)(\mathcal{L}''')^2 c^4}{6(p+2)^2(p+4)} [\tau - 3\nu]^2. \end{aligned} \tag{70}$$

Here, the first term is the square of the average bias, and the second term is "variance of the bias" – note that the second term is proportional to an anisotropy factor $[\tau - 3\nu]^2$ which vanishes for spherically symmetric distance distributions such as Shell and Normal.

The anisotropy factor is nonzero for the the Axis and Bernoulli distributions. The Axis distibution (with only $2p$ directions in p dimensions) has the highest variance-of-bias and is arguably the least reliable. The Bernoulli distribution (with 2^p directions in p dimensions) has a positive variance-of-bias, but for large p the contribution of that variance to the total squared bias scales only as $O(1/p^2)$ and the effect is about a percent when $p = 15$.

It is worth noting that for Bernoulli (and Shell), the *dominant* contribution to the bias scales not with τ, but with ν. In other words, neglecting the effect of cross-derivative terms will lead to completely unreliable conclusions about the behavior of stochastic approximation algorithms on generically oriented loss functions.

III. Numerical Studies

A. Comparison for Chin's loss function

In Chin [5], various finite difference stochastic algorithms are compared using a loss function[7]

$$L(\mathbf{x}) = |\mathbf{x}|^2 + \sum_{i=1}^{p} e^{x_i/p}, \tag{71}$$

which can also be written

$$L(\mathbf{x}) = \sum_{i=1}^{p} \mathcal{L}(x_i), \tag{72}$$

with $\mathcal{L}(x) = x^2 + e^{x/p}$. This is something of a degenerate example; although it is nominally a p-dimensional problem, it is really the same one-dimensional problem, duplicated p times. If such a loss function encountered in practice were known to have this structure, then one could adapt the algorithm to exploit this symmetry. But it is reasonable to imagine that the practitioner is unaware of this structure, and wants to apply a general-purpose optimization algorithm to it. One particular aspect of the symmetry in Eq. (72) is that the cross-derivatives are zero, and so Eq. (22) simplifes in this case to $b_\ell = (1/6)c^2\tau L_{\ell\ell\ell}$ and the bias is simply proportional to the fourth moment τ defined in Eq. (20). As Table I shows, the Bernoulli distribution has the smallest τ. (Chin [5] says that uniform distribution on a spherical shell is less accurate than the

[7] This is Eq. (4.1) from Chin [5], apart from a minor typo in the original.

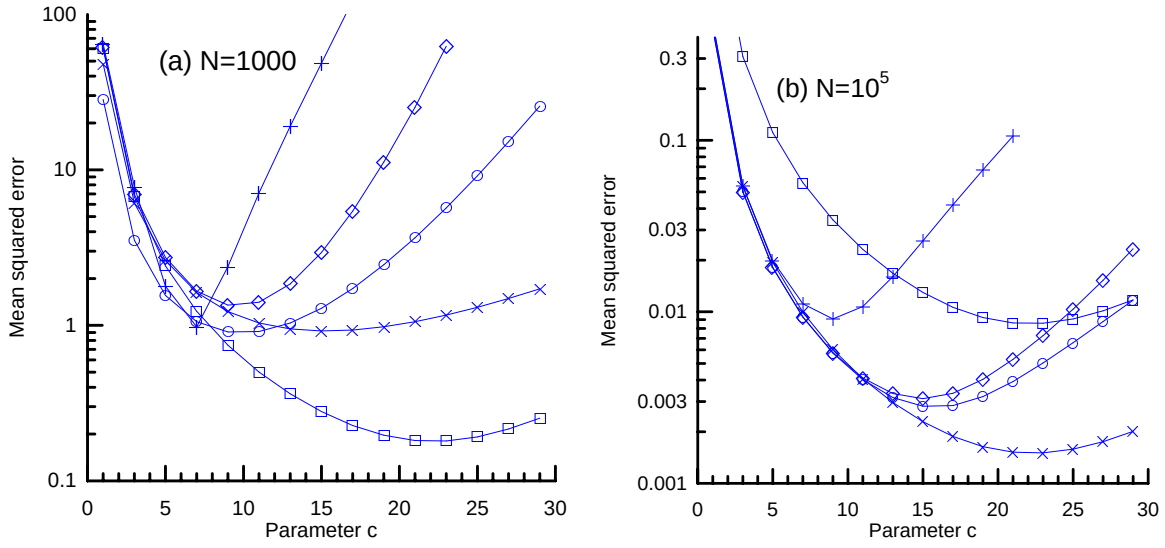

Fig. 1. Mean squared error for different choices of random direction distribution using the Chin loss function in Eq. (71), plotted as a function of the parameter c. Here, plus ($+$) denotes the axis distribution, cross ($\times$) denotes the Bernoulli distribution, circle ($\circ$) denotes the uniform shell distribution, and diamond (3) is the normal distribution. We use squares (2) to indicate the performance of the FDSA algorithm. Results are based on 100 trials. Shown are **(a)** the pre-asymptotic behavior with $N = 1000$ measurements; **(b)** the asymptotic behavior with $N = 10^5$ measurements.

normal distribution, but as Table I shows, the uniform distribution has a slightly *smaller* τ and will be *more* accurate.)

Figs. 1 and 2 show the results of our own numerical experiments with the Chin loss function. For these experiments, we used parameters based on the experiment reported in Chin [5]: $p = 15$, $\alpha = 1$, $a_o = 0.5$, and $\gamma = 1/6$. (Chin [5] reported using $\beta = 1/6$, but we presume that this is a typo, since that would lead to the nonoptimal $\gamma = 5/12$.) Our choices for c_o generally agreed with the values used by Chin [5]; we estimated the optimal c_o numerically for each distribution, and ran each at its optimum. Each evaluation of the loss function included a gaussian noise level of $\sigma = 1.9365$; this corresponds to a variance in the difference of two loss function estimates of $2\sigma^2 = 7.5$. We used $\mathbf{x}_o = [-0.01, \ldots, -0.01]$ for initial conditions, and express squared error relative to this initial condition:

$$\mathrm{Err}^2 = \frac{|\mathbf{x} - \mathbf{x}^*|^2}{|\mathbf{x}_o - \mathbf{x}^*|^2}, \tag{73}$$

where $\mathbf{x}^* = [-0.0332595052015928, \ldots]$ is the minimum of the loss function.

Fig. 1 shows that at the Bernoulli-optimal value of $c_o = 22.5$, the Bernoulli distribution performed dramatically better than the other distributions. But if each distribution is allowed to use its own optimum, the difference is not as dramatic, though still is significant. It is true that

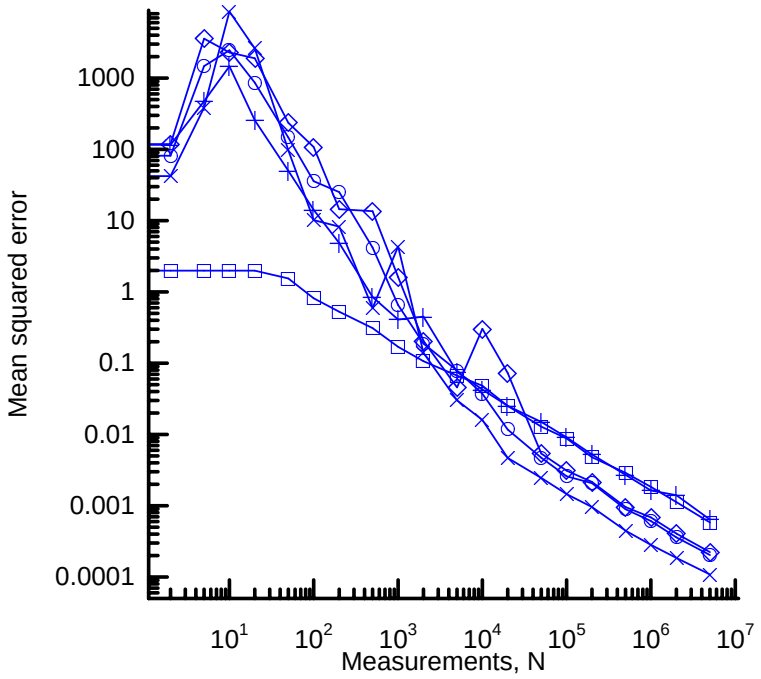

Fig. 2. Mean squared error, based on 10 trials, for different choices of random direction with the Chin loss function in Eq. (71), plotted as a function of the number of measurements, N. A separate optimal value of c_o was chosen for each distribution: $c_o = 9$ for the axis (+) distribution; $c_o = 15$ for the shell (o) and normal (3) distributions; and $c_o = 22.5$ for the Bernoulli ($\times$) distribution; we also used $c_o = 22.5$ for the FDSA (2) algorithm.

one does not generally have access to the optimal c_o in real problems; still, a fair comparison requires that the performance of each method be determined from the c_o that is optimal for that method. Using Eq. (40) as a guide, we expect $c_o \sim |\tilde{\mathbf{b}}|^{-1/3}$.

Since the cross-derivatives vanish for this loss function, we see from Eq. (22) and Eq. (23) that FDSA and Bernoulli have the same bias, and we observe the same optimal c_o for those two algorithms. The scaled bias term $\tilde{\mathbf{b}}$ is p times larger for Axis than Bernoulli, which suggests that the optimal c_o for Axis will be $p^{-1/3}$ times c_o for Bernoulli (that is: about 40% as large for $p = 15$), which is consistent with Fig. 1. The scaled bias term is only about three times larger for Shell and Normal than it is for Bernoulli, which suggests that the optimal c_o for those two algorithms will be about $3^{-1/3} \approx 0.7$ the size of the optimal c_o for Bernoulli.

Fig. 2 shows that the scaling of squared error as $O(N^{-2/3})$ is observed for all the schemes, but coefficient of that scaling is best for Bernoulli, not as good for Shell and Normal, and worst for the Axis distribution and for FDSA. This confirms the theoretical predictions in Chin [5], and agrees with the scaling in Eq. (42) and Eq. (43). We also note that in the asymptotic

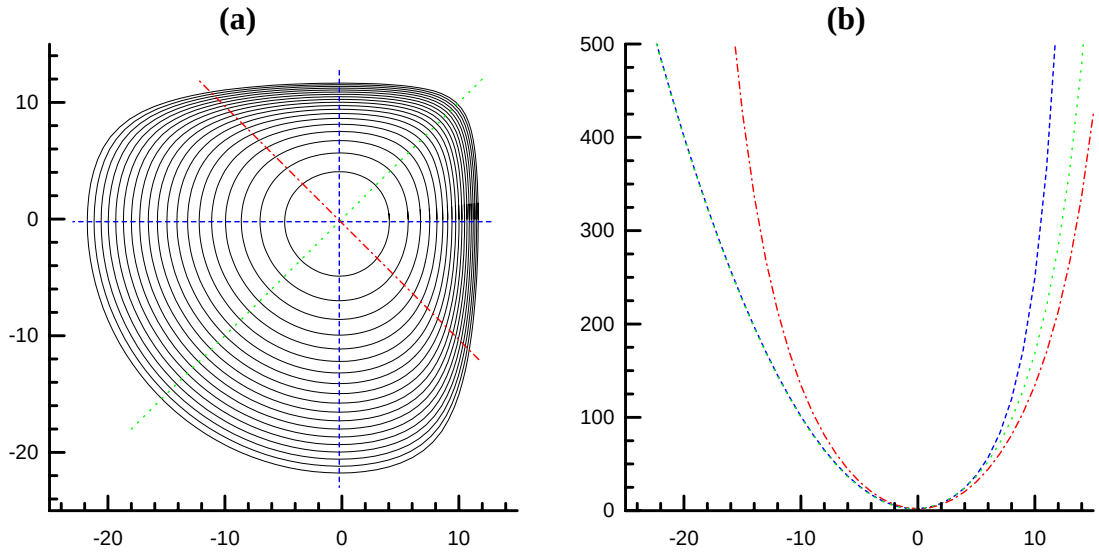

Fig. 3. **(a)** Contour plot of the loss function in Eq. (71) for $p = 2$ dimensions. Contours are shown for every $\Delta L = 25$ up to $L = 500$. The horizontal and vertical dashed lines correspond to the axis-oriented directions; the diagonal dotted and dash-dotted lines correspond to the two Bernoulli directions. **(b)** Loss function is plotted as a function of the position along the slices for the three slices in panel (a). The dash-dotted line is a perfect parabola (third derivative is zero), and departures from this are greatest in the directions parallel to the axes.

regime, Bernoulli achieves the same accuracy as FDSA with $p = 15$ times fewer measurements. Nonetheless, it is interesting to see that for $N \ll 1000$ measurements, the FDSA algorithm is substantially more accurate.

Fig. 3 provides a visualization of these results for the $p = 2$ case. Along the axis directions, much steeper deviations from the limiting quadratic are seen, while the slopes are more benign in the diagonal Bernoulli directions.

B. Another contrived loss function

We investigated a loss function given by

$$L(\mathbf{x}) = |\mathbf{x}|^2 + K\left[\sum_i x_i \sum_i x_i^2 - \sum_i x_i^3\right]/p^2. \tag{74}$$

This is a loss function for which the direct third derivatives $L_{\ell\ell\ell}$ are zero, but the cross-derivatives are nonzero. Where Chin's function depends on τ only, this function depends on ν only. We would therefore expect to achieve the least bias for the Axis distribution; in fact, since $\nu = 0$ for this case, we expect zero bias for this distribution. This is confirmed in Fig. 4(a) which shows

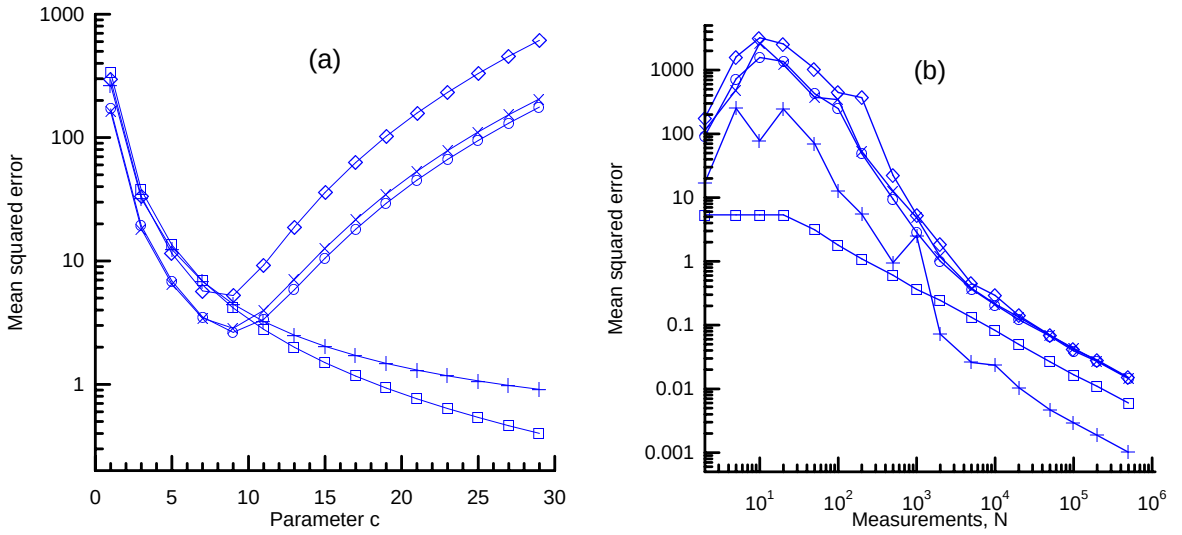

Fig. 4. Mean squared error for different choices of random direction distribution applied to the contrived loss function in Eq. (74) with $K = 0.01$ and $p = 15$. See caption to Fig. 1 for explanation of symbols. **(a)** Plotted against the parameter c, an optimum of $c = 9$ is observed for Bernoulli, Shell, and Normal. For Axis and FDSA, the bias is zero for arbitrarily large c, so the larger c the better. We used $c = 30$ as the "optimum." **(b)** Plotted against the number of measurements N, we see that Axis and FDSA have the best asymptotic performance, while Bernoulli, Normal, and Shell, are worse.

that the performance continues to improve monotonically as c_o is increased for Axis. Since $\nu = 1$ for the Bernoulli and Normal distributions, we expect essentially the same performance for these two. We have $\nu = p/(p+2)$ for Shell, which suggests slightly better performance (compared to Bernoulli and Normal), though for large p, the difference is small.

Note that the empirical comparison of Axis with the FDSA algorithm in Fig. 4 is artificial, since we arbitrarily chose to use the same value of c_o for both methods. The value we chose is not optimal for either method since the performance improves monotonically with increasing c_o for both methods. The superior small-N performance of FDSA is again evident.

C. Rotated Chin function

To investigate the performance of these different random direction schemes on a more generic loss function, we altered Chin's function so that it would not be so neatly aligned with the axes. We chose a random[8] rotation matrix U and defined

$$\tilde{L}(\mathbf{x}) = L(U\mathbf{x}) = |U\mathbf{x}|^2 + \sum_i e^{(U\mathbf{x})_i/p}, \tag{75}$$

[8]We made a matrix with random entries, applied a QR decomposition, and used the orthogonal Q matrix.

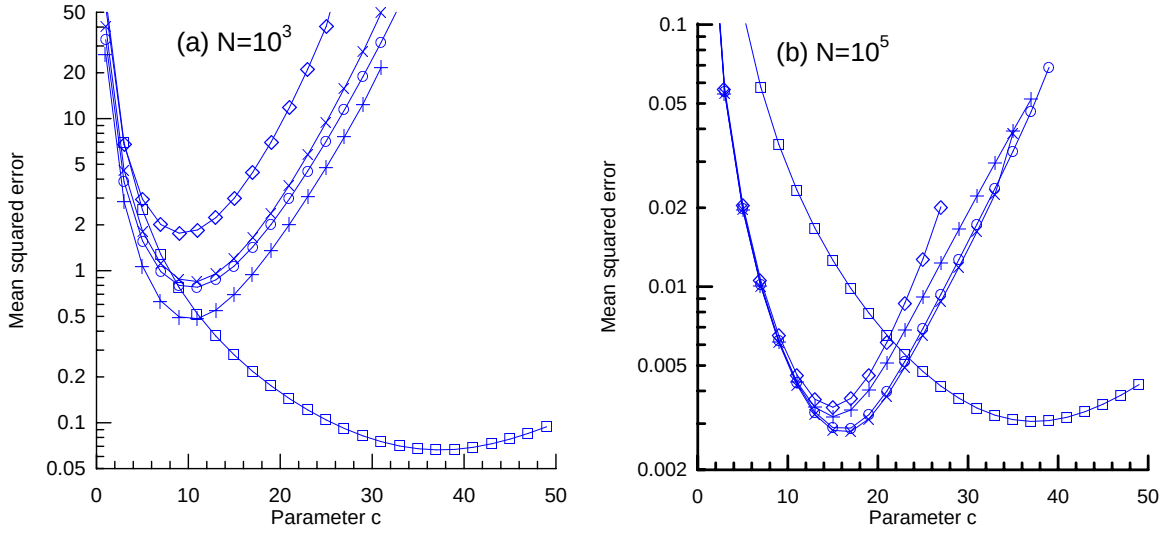

Fig. 5. Mean squared error as a function of the parameter c, using the randomly rotated Chin loss function, defined in Eq. (76), for different choices of random direction distribution. Symbols are defined in the caption to Fig. 1. Shown are **(a)** the pre-asymptotic behavior with $N = 1000$ measurements; **(b)** the asymptotic behavior with $N = 10^5$ measurements.

or

$$\tilde{L}(\mathbf{x}) = \sum_{i=1}^{p} \left[x_i^2 + \exp\left(\sum_j U_{ij} x_j / p \right) \right], \tag{76}$$

which uses $|\mathbf{x}|^2 = |U\mathbf{x}|^2$ since U is orthogonal.

For this loss function, we did the same experiment as for the unrotated Chin loss function, and this time (as shown in Figs. 5 and 6), we found that the asymptotic performance for each of the different random direction distributions was virtually the same. The performance of FDSA was also comparable to the RDSA algorithms, and in the pre-asymptotic regime, FDSA was better. This is in contrast to the oft-made claim that RDSA algorithms are p times more efficient than FDSA "under reasonably general conditions." Note that the optimal c_o for FDSA is substantially larger (by a factor of $p^{1/3} \approx 2.5$ for $p = 15$) than the optimal c_o for the various RDSA algorithms. This follows from the scaling of $c_o \sim |\tilde{\mathbf{b}}|^{-1/3}$ in Eq. (40) and the fact that in the generically rotated case, the FDSA bias is p times smaller than the RDSA bias (compare Eq. (55) and Eq. (54)).

IV. Non-independent choice of direction

In the comparisons so far, we have assumed that random directions were chosen to be independent and identically distributed (IID) according to the distribution of choice. Taking an

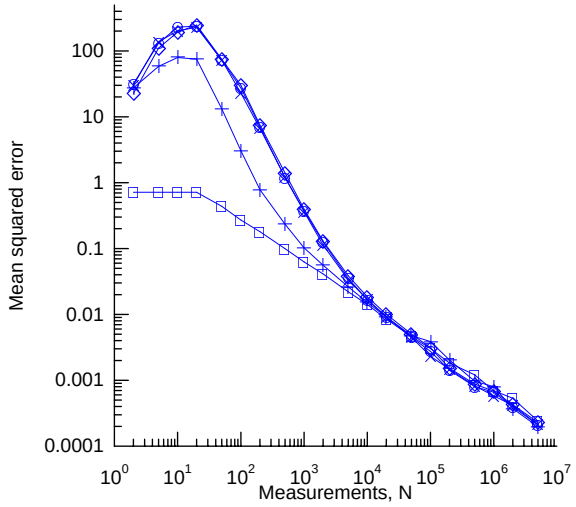

Fig. 6. Mean squared error for the rotated Chin loss function, defined in Eq. (76), plotted against the number N of measurements of the loss function. Symbols are defined in the caption to Fig. 1.

analogy with the use of quasirandom numbers for Monte Carlo integration, we also performed some experiments in which we considered random directions that were not strictly IID. In particular, in our $p = 15$ dimensional space, we took random directions that were constrained to be perpendicular to the P previous random directions where $0 < P < p$.

Interestingly, we found that the asymptotic performance was hardly affected. But Fig. 7 shows that in the pre-asymptotic regime, this "quasirandom" choice of direction seems to provide a very noticeable improvement.

V. Conclusions

The performance of different variants of the RDSA algorithm is driven by the accuracy with which they estimate the gradient of the loss function. In Eq. (22), that bias is expressed in terms of two statistics – τ and ν – which characterize the random direction distribution, and in terms of the third derivatives of the loss function.

For loss functions which happen to be aligned with the coordinate axes (*e.g.*, those which can be expressed as a sum of one-dimensional loss functions of each of the coordinate values) in such a way that their third cross-derivatives are all zero (that is, $\partial^3 L/\partial x_i \partial x_j \partial x_k = 0$ unless $i = j = k$), the bias depends only on τ, and as seen in Table I, τ is smallest for the Bernoulli distribution. For such loss functions, it is also the case that RDSA with a Bernoulli distribution can be up to p times more efficient than the Kiefer-Wolfowitz-Blum FDSA algorithm.

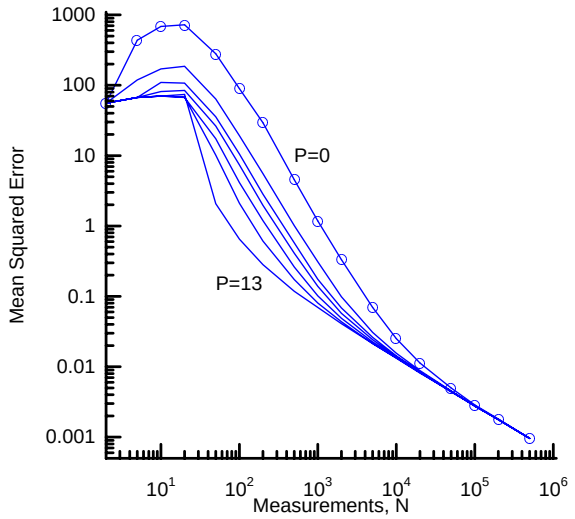

Fig. 7. Mean squared error (over 1000 trials) plotted against number of measurements of Chin's loss function, defined in Eq. (71), for $p = 15$. Here, "quasirandom" directions are chosen from the uniform shell distribution; each new choice of direction is constrained to be perpendicular to the P directions that came before it. The top curve, labelled with open circles (o), corresponds to $P = 0$, and is the the ordinary random distribution. The solid curves correspond to values of $P = 1, 2, 3, 5, 8, 13$, with the larger values of P producing the smaller mean squared errors.

However, this behavior is not generic. We have shown that it is possible to contrive a loss function for which the Bernoulli distribution is *less* efficient than other distributions. And by considering loss functions that are randomly rotated with respect to the coordinate axes, we find that the average bias depends only on the fourth radial moment of the distribution, and is the same for Axis, Shell, and Bernoulli. Furthermore, we find that the FDSA algorithm performs as well as the various RDSA algorithms in the asymptotic limit, and we observe numerically that FDSA is superior to RDSA in the pre-asymptotic regime. Finally, by looking at the squared bias, averaged over random rotations, we find that the optimal choice of distributions in RDSA is given by a uniform spherical shell.

While we believe that it is important to understand the relative behavior of these algorithms on generic loss functions, it is true that in practice, one seldom needs to optimize a generic loss function, but instead needs to optimize a particular loss function associated with an actual problem at hand. It is not unreasonable to imagine that this loss function, expressed in its "natural" or "physical" coordinates, may have unusually small cross-derivatives compared to a generic function. In this case, a Bernoulli distribution will be advantageous. If this judgement turns out to be incorrect, and the loss function is not fortuitously aligned with the coordinate axes, then it

is worth remarking that the Bernoulli distribution is not far from optimal, particularly in high dimensional spaces. Thus, while we vigorously reject some of the mistaken theoretical arguments that have appeared in the literature, regarding the general superiority of the Bernoulli distribution over other choices, we feel that in practical application, it is a very reasonable choice.

But we should add that in practical situations, other practical considerations may be more important. In the absence of noise, convergence can be exponentially fast [14], and if the noise is small, it may be more useful to employ a "search then converge" approach in which a more direct gradient descent is used to approach the general vicinity of $\mathbf{x}^*$, and then stochastic approximation is used to narrow in on the optimum [15], [16]. A direct search, based on the Nelder-Mead simplex algorithm, has also been proposed, in which multiple measurements are used to reduce the measurement noise as the optimum is approached [17]. Another impracticality with the standard formulation is that optimal parameters a_o and c_o depend on properties of the loss function which are not usually available; variants of SA algorithms which provide adaptive estimates of these parameters should be considered [16], [18], including iterate averaging methods [19]. Approaches using quasirandom or "common" random numbers [20] may improve convergence. Finally, we remark that while asymptotic behavior is an important guide to performance, good practical algorithms will have to behave well in the pre-asymptotic regime as well. A method called "retrospective" approximation [21] claims that in exchange for a somewhat larger asymptotic variance, a more robust convergence can be achieved. Recent texts [22], [23], [24] provide an overview of some of these issues in the context of some useful applications.

APPENDIX I
MOMENTS OF A UNIT SPHERICAL SHELL

In this appendix, we derive an expression for the moments of the isotropic distribution of $(x_1, \ldots, x_p)$ constrained by $x_1^2 + \ldots + x_p^2 = 1$. Define

$$F_{p,\mathbf{m}} \equiv E\Big\{ x_1^{m_1} x_2^{m_2} \cdots x_p^{m_p} \Big\} = \frac{\int_{\mathcal{S}} x_1^{m_1} x_2^{m_2} \cdots x_p^{m_p} dS}{\int_{\mathcal{S}} dS}, \tag{77}$$

where $\int_{\mathcal{S}} \bullet \, dS$ corresponds to the integral over the surface of the unit sphere.

Let $\mathbf{z}$ be a p-dimensional vector in which each component is independently distributed as $\mathcal{N}(0,1)$. We will compute the quantity $E\Big\{ z_1^{m_1} \cdots z_p^{m_p} \Big\}$ in two different ways, and by equating the results, we will obtain an expression for $F_{p,\mathbf{m}}$.

For our first computation, we use the fact that the gaussian components are independent, and write

$$E\Big\{\, z_1^{m_1} \cdots z_p^{m_p} \,\Big\} = E\{\, z_1^{m_1} \,\} \cdots E\Big\{\, z_p^{m_p} \,\Big\}. \tag{78}$$

We have for each component that

$$E\{\, z^m \,\} = \frac{\int_{-\infty}^{\infty} z^m e^{-z^2/2}\, dz}{\int_{-\infty}^{\infty} e^{-z^2/2}\, dz} = (m-1)!! \tag{79}$$

if m is even (and zero if m is odd), and so

$$E\Big\{\, z_1^{m_1} \cdots z_p^{m_p} \,\Big\} = (m_1 - 1)!! \cdots (m_p - 1)!! \tag{80}$$

again under the assumption that every one of the components $m_1, \ldots, m_p$ are even.

For our second derivation, we use the relation $\mathbf{z} = r\mathbf{x}$, where r is the radius of the vector $\mathbf{z}$, and $\mathbf{x}$ is the unit vector in the direction of $\mathbf{z}$. Thus we have

$$z_1^{m_1} \cdots z_p^{m_p} = r^{m_1 + \cdots + m_p} x_1^{m_1} \cdots x_p^{m_p}. \tag{81}$$

Rather than integrate coordinate-wise, as we did in Eq. (80), we will use a kind of polar coordinates, and write

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \bullet \; dz_1 \, \ldots \, dz_p = \int_0^{\infty} \int_{\mathcal{S}} \bullet \; r^{p-1}\, dr\, dS. \tag{82}$$

Then,

$$\begin{aligned}
E\Big\{\, z_1^{m_1} \cdots z_p^{m_p} \,\Big\} &= \frac{\int_0^{\infty} \int_{\mathcal{S}} z_1^{m_1} \cdots z_p^{m_p}\, e^{-r^2/2}\, r^{p-1}\, dr\, dS}{\int_0^{\infty} \int_{\mathcal{S}} e^{-r^2/2}\, r^{p-1}\, dr\, dS} && (83)\\
&= \frac{\Big(\int_0^{\infty} e^{-r^2/2}\, r^{m_1 + \cdots + m_p}\, r^{p-1}\, dr\Big)\Big(\int_{\mathcal{S}} x_1^{m_1} \cdots x_p^{m_p}\, dS\Big)}{\int_0^{\infty} e^{-r^2/2}\, r^{p-1}\, dr \;\int_{\mathcal{S}}\, dS} && (84)\\
&= \frac{(p + m_1 + \cdots + m_p - 2)!!\, F_{p,\mathbf{m}}}{(p-2)!!}. && (85)
\end{aligned}$$

Comparing Eq. (80) with Eq. (85), we have

$$F_{p,\mathbf{m}} = \frac{(m_1 - 1)!! \cdots (m_p - 1)!! (p-2)!!}{(p + m_1 + \cdots + m_p - 2)!!}, \tag{86}$$

which can also be written

$$E\Big\{\, x_1^{m_1} x_2^{m_2} \cdots x_p^{m_p} \,\Big\} = \frac{(m_1 - 1)!! \cdots (m_p - 1)!!}{p(p+2) \cdots (p + m_1 + m_2 + \cdots + m_p - 2)}. \tag{87}$$

As special cases of Eq. (87), we have:

$$E\Big\{ x_i^2 \Big\} = \frac{1}{p}, \tag{88}$$

$$E\Big\{ x_i^4 \Big\} = \frac{3}{p(p+2)}, \tag{89}$$

$$E\Big\{ x_i^2 x_j^2 \Big\} = \frac{1}{p(p+2)}, \tag{90}$$

$$E\Big\{ x_i^6 \Big\} = \frac{15}{p(p+2)(p+4)}, \tag{91}$$

$$E\Big\{ x_i^4 x_j^2 \Big\} = \frac{3}{p(p+2)(p+4)}, \tag{92}$$

$$E\Big\{ x_i^2 x_j^2 x_k^2 \Big\} = \frac{1}{p(p+2)(p+4)}. \tag{93}$$

APPENDIX II
HIGHER-ORDER GRADIENT APPROXIMATIONS

The derivations in the text of this article are based on two-point gradient estimators, but many of the same conclusions apply to higher-order estimators as well. In this appendix, we will describe these higher-order estimators, and how the results in the text would be modified for that case.

Fabian [25], [26] described a modification of the FDSA algorithm that used $2sp$ loss function evaluations (instead of $2p$) to achieve a gradient estimate with much smaller bias. Here, the gradient of the ith coordinate is given by

$$\hat{g}_i(\mathbf{x}) = \frac{\sum_{j=1}^{s} v_j \left[\hat{L}(\mathbf{x} + cu_j\mathbf{e}_i) - \hat{L}(\mathbf{x} - cu_j\mathbf{e}_i) \right]}{2c}, \tag{94}$$

where the constants u_j, v_j are chosen beforehand to satisfy

$$\begin{aligned} \sum_j v_j u_j &= 1, \\ \sum_j v_j u_j^3 &= 0, \\ &\vdots \\ \sum_j v_j u_j^{2s-1} &= 0. \end{aligned} \tag{95}$$

With these conditions, the bias will scale like $O(c^{2s})$. A smaller bias means that the schedule for reducing $c_k \to 0$ can go more slowly, and in fact, by taking $c_k \sim k^{1/(4s+2)}$ we can achive

$\text{Err}^2 \sim k^{-2s/(2s+1)}$, which for large enough s approaches the Robbins-Monro scaling of $O(k^{-1})$. Polyak and Tsybakov [27] described a randomized variant of this approach which can achieve the same $k^{-2s/(2s+1)}$ scaling with only two (or even one!) measurement of the loss function for each iteration. The practical utility of this approach, however, remains to be demonstrated.

A. Variance of a higher-order estimator

If we use the higher-order estimator of gradient in Eq. (94), we find that the noise the ith component of the gradient is given by

$$\eta_i = \frac{\sum_j v_j(\epsilon_+^j - \epsilon_-^j)}{2c}, \tag{96}$$

and in particular, we can compute the covariance and variance.

$$E\Big\{ \boldsymbol{\eta}\boldsymbol{\eta}^T \Big\} = \frac{\sigma^2 \sum_j v_j^2}{2c^2} I, \tag{97}$$

$$E\Big\{ \boldsymbol{\eta}^T\boldsymbol{\eta} \Big\} = \frac{p\sigma^2 \sum_j v_j^2}{2c^2}. \tag{98}$$

Although we have not seen it suggested in the literature, it is straightforward to apply Fabian's higher-order gradient estimators to the RDSA algorithms as well as to the FDSA algorithm. That is:

$$\hat{\mathbf{g}}(\mathbf{x}) = \frac{\sum_{j=1}^s v_j \left[\hat{L}(\mathbf{x} + cu_j\boldsymbol{\xi}) - \hat{L}(\mathbf{x} - cu_j\boldsymbol{\xi})\right]}{2c}\boldsymbol{\xi}. \tag{99}$$

Just as we saw with the two-point gradient estimators, the covariance and variance are the same for all distributions of RDSA, and are the same as for FDSA. That is, the variance of the estimator in Eq. (99) is given by Eq. (98).

In the spirit of Polyak and Tsybakov [27], we can consider replacing the coefficients v_j with random variables q_j that have the property $E\{ q_j \} = v_j$. For instance, we can randomly choose $k \in \{1, \ldots, s\}$ with probability p_k, and set $q_j = 0$ for $j \neq k$ and $q_k = v_k/p_k$. This way, q_j is zero for all but one value of j, which means that the higher-order gradient estimator can be computed with only two computations of the loss function. For this randomized variant, $E\Big\{ \boldsymbol{\eta}^T\boldsymbol{\eta} \Big\} = \dfrac{p\sigma^2}{2c^2} \displaystyle\sum_j E\Big\{ q_j^2 \Big\}$, which is generally a larger variance than Eq. (98). The optimal design uses $p_k = |v_k| / \sum_k |v_k|$, and in that case the variance is given by

$$E\Big\{ \boldsymbol{\eta}^T\boldsymbol{\eta} \Big\} = \frac{p\sigma^2}{2c^2} \left(\sum_j |v_j| \right)^2. \tag{100}$$

But again, the practical utility of this approach has not been investigated.

	$E\{\,\xi_i^6\,\}$	$E\left\{\,\xi_i^4\xi_j^2\,\right\}$	$E\left\{\,\xi_i^2\xi_j^2\xi_k^2\,\right\}$
Distribution			
Bernoulli	1	1	1
Axis	p^2	0	0
Normal	15	3	1
Shell	$15p^2/(p+2)(p+4)$	$3p^2/(p+2)(p+4)$	$p^2/(p+2)(p+4)$

TABLE II

FOUR RANDOM DIRECTION DISTRIBUTIONS AND STATISTICS THAT CHARACTERIZE THE BIAS IN THE HIGHER ORDER ($s = 2$) GRADIENT ESTIMATOR.

B. Bias for higher-order gradient estimation

Using Fabian's higher-order estimator of FDSA gradient in Eq. (94), it is fairly straightforward to write the bias

$$b_\ell = \frac{c^{2s}}{(2s+1)!}\left(\sum_j v_j u_j^{2s+1}\right)\frac{\partial^{2s+1}L}{\partial x_\ell^{2s+1}}. \tag{101}$$

The case for RDSA is a little more complicated, but it is not too hard to show that Eq. (99) leads to

$$\mathbf{b} = \frac{c^{2s}}{(2s+1)!}\left(\sum_j v_j u_j^{2s+1}\right) E\left\{\, L_\xi^{(2s+1)}\boldsymbol{\xi}\,\right\}. \tag{102}$$

Considering in particular the $s = 2$ estimator, we have for the ℓth component of the derivative:

$$\left(L_\xi^{(5)}\boldsymbol{\xi}\right)_\ell = \sum_{ijkmn} L_{ijkmn}\xi_i\xi_j\xi_k\xi_m\xi_n\xi_\ell. \tag{103}$$

When we take expectation value, we only keep the even orders. In the two-point gradient estimator, this led to a pair of statistics – τ and ν – for characterizing the different RDSA distributions. For this higher-order estimator, we now have three such statistics: $E\{\,\xi_i^6\,\}$, $E\left\{\,\xi_i^4\xi_j^2\,\right\}$, and $E\left\{\,\xi_i^2\xi_j^2\xi_k^2\,\right\}$. Table II shows how these statistics vary with the choice of random direction distribution.

Again invoking smoothness in the loss function L, we note that the five combinations of $L_{iiii\ell}$ are equal (namely: $L_{iiii\ell} = L_{iii\ell i} = L_{ii\ell ii} = L_{i\ell iii} = L_{\ell iiii}$), and similarly for the ten combinations of $L_{ii\ell\ell\ell}$ and the thirty combinations of $L_{iijj\ell}$. This allows us to write the analog

of Eq. (22) for the higher-order gradient estimator:

$$b_\ell = \frac{c^4}{120}\left(\sum_j v_j u_j^5\right)\Bigg[\quad E\Big\{\,\xi_i^6\,\Big\}\,L_{\ell\ell\ell\ell\ell\ell} + 5E\Big\{\,\xi_i^4\xi_j^2\,\Big\}\sum_{i\neq\ell} L_{iiii\ell} + 10E\Big\{\,\xi_i^4\xi_j^2\,\Big\}\sum_{i\neq\ell} L_{ii\ell\ell\ell} + 30E\Big\{\,\xi_i^2\xi_j^2\xi_k^2\,\Big\}\sum_{i\neq j\neq\ell} L_{iijj\ell}\Bigg]. \quad (104)$$

In the special case that the cross-derivatives for the loss function L are zero, then – as was the case with the two-point gradient estimator – the best choice of random directions is given by the Bernoulli distribution. It is evident in Eq. (104) that the relevant statistic in that case is $E\{\,\xi_i^6\,\}$, and Table II shows that the Bernoulli distribution exhibits the smallest value.

C. Rotation-averaged bias for higher-order gradient estimators

In the more generic case that the loss function and the coordinate axes are not aligned in any particular way, then we can consider the rotation-averaged bias as an indicator of performance. In this case the statistics of interest are $\langle E\{\,(U\boldsymbol{\xi})_i^6\,\}\rangle$, $\Big\langle E\Big\{\,(U\boldsymbol{\xi})_i^4(U\boldsymbol{\xi})_j^2\,\Big\}\Big\rangle$, and $\Big\langle E\Big\{\,(U\boldsymbol{\xi})_i^2(U\boldsymbol{\xi})_j^2(U\boldsymbol{\xi})_k^2\,\Big\}\Big\rangle$. For a particular rotation U, we have

$$E\Big\{\,(U\boldsymbol{\xi})_\ell^6\,\Big\} = \sum_{ijkmno} u_{i\ell}u_{j\ell}u_{k\ell}u_{m\ell}u_{n\ell}u_{o\ell}E\{\,\xi_i\xi_j\xi_k\xi_m\xi_n\xi_o\,\}, \quad (105)$$

and keeping only the even moments,

$$E\Big\{\,(U\boldsymbol{\xi})_\ell^6\,\Big\} = E\Big\{\,\xi_i^6\,\Big\}\sum_i u_{i\ell}^6 + 15E\Big\{\,\xi_i^4\xi_j^2\,\Big\}\sum_{i\neq j} u_{i\ell}^4u_{j\ell}^2 + 90E\Big\{\,\xi_i^2\xi_j^2\xi_k^2\,\Big\}\sum_{i\neq j\neq k} u_{i\ell}^2u_{j\ell}^2u_{k\ell}^2, \quad (106)$$

where the coefficients 1, 15, and 90 come from $6!/6!$, $6!/(4!2!)$, and $6!/(2!2!2!)$ respectively. Taking the average over all rotations,

$$\begin{aligned}\Big\langle E\Big\{\,(U\boldsymbol{\xi})_\ell^6\,\Big\}\Big\rangle &= E\Big\{\,\xi_i^6\,\Big\}p\left\langle u_{i\ell}^6\right\rangle + 15E\Big\{\,\xi_i^4\xi_j^2\,\Big\}p(p-1)\left\langle u_{i\ell}^4u_{j\ell}^2\right\rangle + \\ &\quad 90E\Big\{\,\xi_i^2\xi_j^2\xi_k^2\,\Big\}p(p-1)(p-2)\left\langle u_{i\ell}^2u_{j\ell}^2u_{k\ell}^2\right\rangle \quad (107)\\ &= E\Big\{\,\xi_i^6\,\Big\}\frac{15p}{(p+2)(p+4)} + 15E\Big\{\,\xi_i^4\xi_j^2\,\Big\}\frac{3p(p-1)}{(p+2)(p+4)} + \\ &\quad 90E\Big\{\,\xi_i^2\xi_j^2\xi_k^2\,\Big\}\frac{p(p-1)(p-2)}{(p+2)(p+4)} \quad (108)\end{aligned}$$

However, note that the sixth radial moment is given by

$$\begin{aligned}E\Big\{\,r^6\,\Big\} &= E\left\{\left(\sum_i x_i^2\right)^3\right\} = E\left\{\sum_{ijk} x_i^2x_j^2x_k^2\right\}\\ &= pE\Big\{\,\xi_i^6\,\Big\} + 3p(p-1)E\Big\{\,\xi_i^4\xi_j^2\,\Big\} + 6p(p-1)(p-2)E\Big\{\,\xi_i^2\xi_j^2\xi_k^2\,\Big\}, \quad (109)\end{aligned}$$

where the coefficients 1, 3, and 6 come from 3!/3!, 3!/(2!1!), and 3!/(1!1!1!). Thus, we can express

$$\Big\langle E\Big\{ (U\boldsymbol{\xi})_\ell^6 \Big\}\Big\rangle = \frac{15}{(p+2)(p+4)} E\Big\{ r^6 \Big\}, \tag{110}$$

and similarly

$$\Big\langle E\Big\{ (U\boldsymbol{\xi})_i^4 (U\boldsymbol{\xi})_j^2 \Big\}\Big\rangle = \frac{3}{(p+2)(p+4)} E\Big\{ r^6 \Big\}, \tag{111}$$

$$\Big\langle E\Big\{ (U\boldsymbol{\xi})_i^2 (U\boldsymbol{\xi})_j^2 (U\boldsymbol{\xi})_k^2 \Big\}\Big\rangle = \frac{1}{(p+2)(p+4)} E\Big\{ r^6 \Big\}. \tag{112}$$

This says that the rotation-averaged bias for the higher-order bias estimator depends only on the radial sixth moment of the direction distribution. That is, like the two-point gradient estimator, the higher-order estimator leads to a bias on generically rotated loss functions which is the same for the Bernoulli, Axis, and Shell distributions of random directions.

We speculate, by analogy with the result in Eq. (70) for two-point gradient estimators, and on the basis of symmetry arguments, that the rotation-averaged squared bias would be smallest for the Shell distribution.

REFERENCES

[1] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Stat.*, vol. 22, pp. 400–407, 1951.

[2] J. R. Blum, "Multidimensional stochastic approximation methods," *Ann. Math. Stat.*, vol. 25, pp. 737–744, 1954.

[3] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *Ann. Math. Stat.*, vol. 23, pp. 462–466, 1952.

[4] H. J. Kushner and D. S. Clark, *Stochastic Approximation for Constrained and Unconstrained Systems*. New York: Springer-Verlag, 1978.

[5] D. C. Chin, "Comparative study of stochastic algorithms for system optimization based on gradient approximation," *IEEE Trans. Systems Man and Cybernetics – Part B*, vol. 27, pp. 244–249, 1997.

[6] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. Automatic Control*, vol. 37, pp. 332–341, 1992.

[7] P. Sadegh and J. C. Spall, "Optimal random perturbations for stochastic approximation using a simultaneous perturbation gradient approximation," *Proc. American Control Conference, 4-6 June 1997, Albuquerque, NM*, pp. 3582–3586, 1997.

[8] I.-J. Wang and E. K. P. Chong, "A deterministic analysis of stochastic approximation with randomized directions," *IEEE Trans. Automatic Control*, vol. 43, pp. 1745–1749, 1998.

[9] J. C. Spall, "Implementation of the simultaneous perturbation algorithm for stochastic optimization," *IEEE Trans. Aerospace and Electronic Systems*, vol. 34, pp. 817–823, 1998.

[10] J. C. Spall, "An overview of the simultaneous perturbation method for efficient optimization," *Johns Hopkins APL Technical Digest*, vol. 19, pp. 482–492, 1998.

[11] J. C. Spall, "Stochastic optimization, stochastic approximation, and simulated annealing," in *Wiley Encyclopedia of Electrical and Electronics Engineering* (J. G. Webster, ed.), vol. 20, pp. 529–542, New York: John Wiley & Sons, 1998.

[12] J. C. Spall. http://www.jhuapl.edu/SPSA/index.html.

[13] V. Fabian, "On asymptotic normality in stochastic approximation," *Ann. Math. Stat.*, vol. 39, pp. 1327–1332, 1968.

[14] L. Gerencsér and Z. Vágó, "The mathematics of noise-free SPSA," *Proc. IEEE Conference on Decision and Control, 4-7 December 2001, Orlando, FL*, pp. 4400–4405, 2001.

[15] C. Darken and J. Moody, "Note on learning rate schedules for stochastic optimization," *Advances in Neural Information Processing Systems*, vol. 3, pp. 832–838, 1990.

[16] C. Darken and J. Moody, "Towards faster stochastic gradient descent," *Advances in Neural Information Processing Systems*, vol. 4, pp. 1009–1016, 1992.

[17] E. Anderson and M. Ferris, "A direct search algorithm for optimization with noisy function evaluations," *SIAM J. Optim.*, vol. 11, pp. 837–857, 2000.

[18] J. C. Spall, "Adaptive stochastic approximation by the simultaneous pertubation method," *IEEE Trans. Automatic Control*, vol. 45, pp. 1839–1853, 2000.

[19] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM J. Control Optim.*, vol. 41, pp. 1404–1409, 1992.

[20] N. L. Kleinman, J. C. Spall, and D. Q. Naiman, "Simulation-based optimization with stochastic approximation using common random numbers," *Management Science*, vol. 45, pp. 1570–1578, 1999.

[21] H. Chen and B. Schmeiser, "Retrospective approximation algorithms for stochastic root finding," in *Proceedings of the 1994 Winter Simulation Conference* (J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila, eds.), 1994.

[22] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. New York: Springer-Verlag, 1997.

[23] H.-F. Chen, *Stochastic Approximation and Its Applications*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 2002.

[24] J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. New York: John Wiley and Sons, 2003.

[25] V. Fabian, "Stochastic approximation of minima with improved asymptotic speed," *Ann. Math. Stat.*, vol. 38, pp. 191–200, 1967.

[26] V. Fabian, "On the choice of design in stochastic approximation methods," *Ann. Math. Stat.*, vol. 39, pp. 457–465, 1968.

[27] B. T. Polyak and A. B. Tsybakov, "Optimal order of accuracy of search algorithms in stochastic optimization," *Problemy Peredachi Informatsii*, vol. 26, pp. 45–53, 1990. [in Russian; English translation in Problems Inform. Transmission, vol. 26, pp. 126–133, 1990].